# Improving the Performance of PC-Based Controllers for Robot-Assisted Surgery

**Claudio Casadei†, Paolo Fiorini‡, Sandra Martelli†, and Marco Montanari†**

†Laboratorio di Biomeccanica
Istituti Ortopedici Rizzoli
Bologna, Italia 40136

‡Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

## Abstract

*This paper describes a PC-based controller for robot-assisted minimally invasive surgery, capable of accurate execution of preoperative plans under X-ray monitoring. The prototype of a recently developed surgical robot is enhanced with a hybrid, position and compliance, control algorithm enabling manual operation and obstacle detection. In an experimental procedure under development, the robot is used as the surgical instrument, moved initially by the surgeon under position accommodation control, and then under hybrid control tracking the planned trajectory. The surgical robot is a clean-room PUMA 260 manipulator interfaced to a personal computer replacing the VAL controller. The control cycle is reduced to 8 ms, thus significantly improving the original robot performance. Preliminary tests have shown good results under position and hybrid control, and the system readiness for more realistic experiments using biological samples.*

## 1. Introduction

Minimally invasive surgery is a prime candidate for the development of robot-assisted procedures that could improve treatment quality, medical personnel safety, and reduce the overall surgery cost. For example, lumbar radiculopathy due to herniated disks can be effectively treated using percutaneous discectomy, a minimally invasive surgery that requires the precise positioning of a guide needle against the lesion. This procedure can be improved by using a robotic arm because of the simplicity of the surgical gesture required. Percutaneous discectomy consists of the removal of the prolapsed tissue by inserting the surgical tools across a small skin incision until they reach the herniated disk. The tools are guided by a needle initially positioned by the surgeon. The needle is inserted into the patient following a linear trajectory, whose length and orientation can be precisely computed during pre-operative planning.

By equipping a small robot with a surgical needle, a robot-assisted discectomy consisting of three main phases is proposed in [9]. In the first phase, the surgeon would position the tip of the needle at the starting point of the insertion, by guiding the robot as if it were a passive mechanical device. During the second phase, the manipulator would autonomously assume the planned needle orientation, and would initiate the motion towards the prolaxed tissue. Finally, upon reaching the target area, the robot would stop and the surgeon would continue the procedure by hand. This approach would increase the precision of the operation, and provide additional safety for the medical personnel. During needle insertion in fact, the surgeon normally relies on fluoroscopic images and patient feedback to avoid damaging nerves and blood vessels. During a robot-assisted surgery, the needle motion can monitored remotely by the surgeon in a location protected by the X-rays, as shown in Figure reffig:puma.

In spite of the need for robot-assisted surgery, very few commercial robots have the performance needed by surgical operations, and none has a cost compatible with the budget of a research laboratory. In the past, researchers have proposed methods to improve the performance of commercial robotic manipulators, which would eventually lead to their use in surgical procedures. An earlier approach is documented in [1, 11], and is based on reducing the control cycle of a standard PUMA manipulator, to improve position and trajectory tracking control. Successive work has focused on making the PUMA manipulators more accessible to programmers, by developing the Robot Control C Library (RCCL) for Unix workstations [5, 7, 8]. The increased computation power of economic personal computers (PC) allows now to use this technology for controlling a manipulator. For example, a PC-based system connected to a robot controller via a serial interface is presented in [10]. Using this system, the operator can select a specific control law, set the controller gains, and compile the control law into a program to be downloaded to the robot controller. In [4] a PC-based controller for a dexterous manipulator is described, which integrates

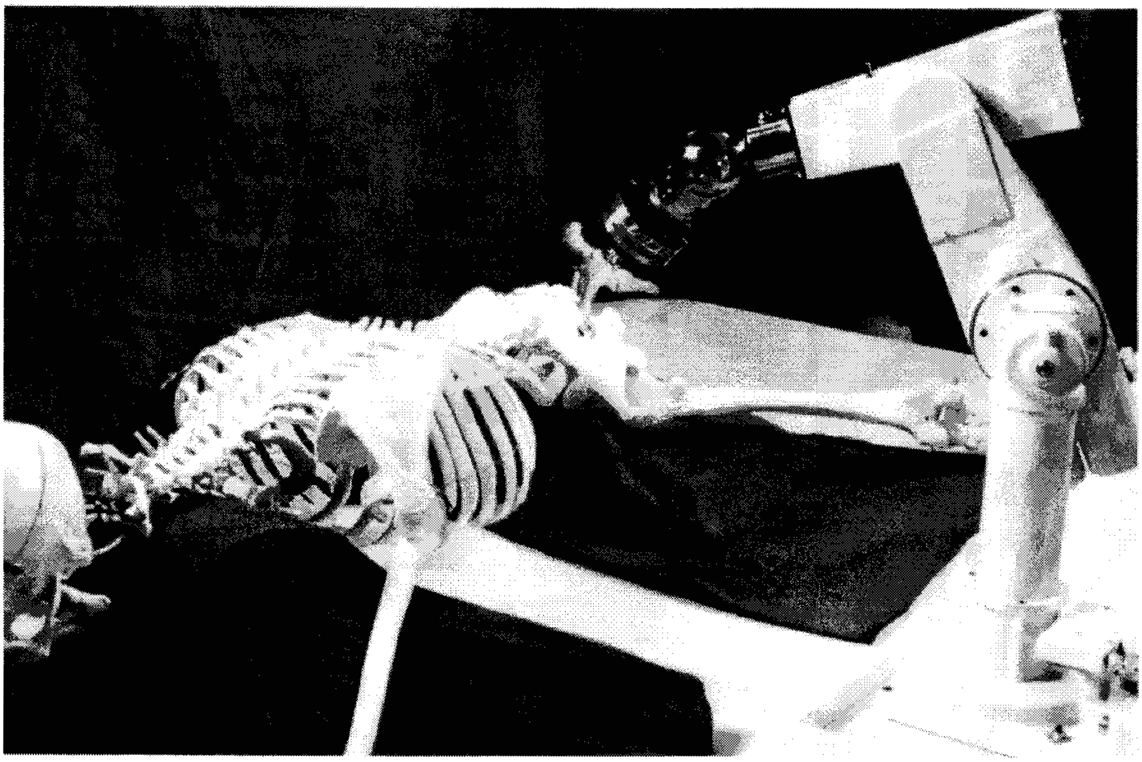

**Figure 1:** The PC-controlled surgical PUMA.

in a single off-the-shelf PC robot control and operator interface functions. A cost effective solution, specifically designed for minimally invasive surgeries is described in [2], and it consists of a PC interfaced to a PUMA 260 using simple hardware enhancements, and parts of the public domain software library RCCL.

This paper describes the enhancements made to the system of [2] to allow position and compliance control of the robot motions, combined in a hybrid controller [12] located in the PC. The paper is organized as follows. Next Section gives a brief description of the system architecture. In Section 3, we summarize the features of the position and compliance control algorithms used. Section 4 describes our current experiments of hybrid control. Finally, in Section 5 we summarize this work and present our future research plans.

## 2. The System Architecture

For completeness, this section briefly describes the main components of the PC-based PUMA controller [2]. The architecture design is driven by the need of interfacing standard laboratory equipment to the PUMA family of manipulators. The hardware of the PC-based surgical workstation consists of a PUMA 260 manipulator, a personal computer, an Assurance Technology Inc. (ATI) 6-axis Force-Torque (FT) sensor, and a solid state TV camera, as shown schematically in Figure 1. The PUMA is interfaced to the PC via serial and parallel ports. The serial port is used for the initial set-up and for downloading the software resident in the PUMA controller. The parallel port is used during normal operation, to exchange data between the PC and the PUMA servo controllers. The PC first down-loads a program called *moper* to the PUMA controller box, and then initiates the control/communication cycle, by starting moper. This program replaces the trajectory generator in the original VAL controller, and acts as a data acquisition and dispatcher for the PUMA joint controllers, exchanging position and set point data with a trajectory generator located in the PC.

The PC software is implemented using the DOS operating system, and is organized in the following threads:
(*i*) An *interrupt handler* activated by the real-time clock of the CMOS memory at $125\,\mu s$ interval.
(*ii*) A *main* program for the housekeeping functions.
(*iii*) A *graphical user interface* for command input.
(*iv*) The *trajectory controller* activated every $8\,ms$.
(*v*) The *FT sensor acquisition.*

The interaction among the tasks is schematically

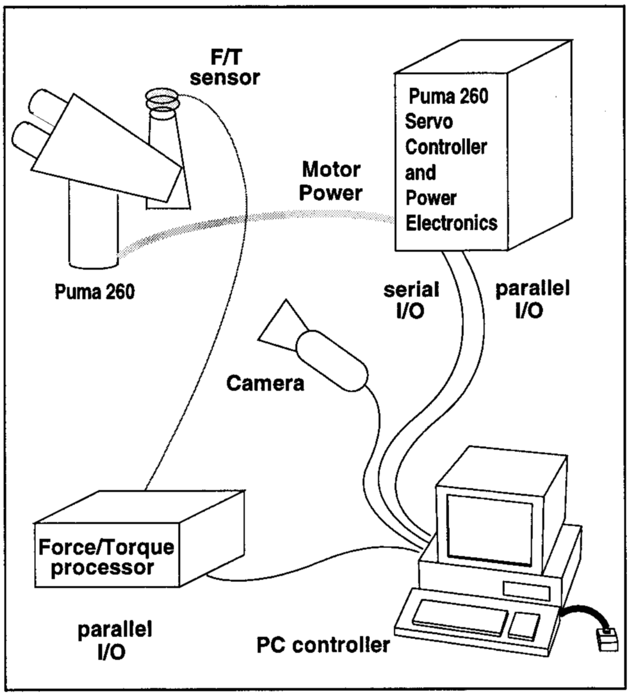


**Figure 2:** Schematic layout of the PC controller.

represented in Figure 2, where the program flow is shown by the arrows connecting the various threads. The controller software starts with the main task ($a$), followed by the user interface ($b$). The clocks ($d$) and ($e$), the control ($c$) and the FT acquisition ($f$) threads are all started by a new motion command. The trajectory is finished at ($g$), when the clocks are turned off and the user interface is enabled again. The FT sensor processor acquires the eight strain-gauge values in approximately $2\,ms$, and transmits them to the PC in $250\,\mu s$. The communication between the PC and the PUMA takes about $4\,ms$, thus leaving another $4\,ms$ for the control algorithms.

## 3. The Control System

The manipulator control system consists of a trajectory generator, located on the PC, where the trajectory set-points are entered, and the joint servo controllers, located in the PUMA electronic box. The connection between the two programs is provided by the communication program *moper*, resident in the PUMA, which collects data from the PUMA joint controllers and exchanges data with the trajectory generator located in the PC. The ability to enhance the PUMA manipulator with more advanced features, such as force and compliance control, hinges on the possibility of reducing the control cycle of the PUMA joint controllers, from its original $28\,ms$ to a shorter cycle suitable for force control. The joint controllers

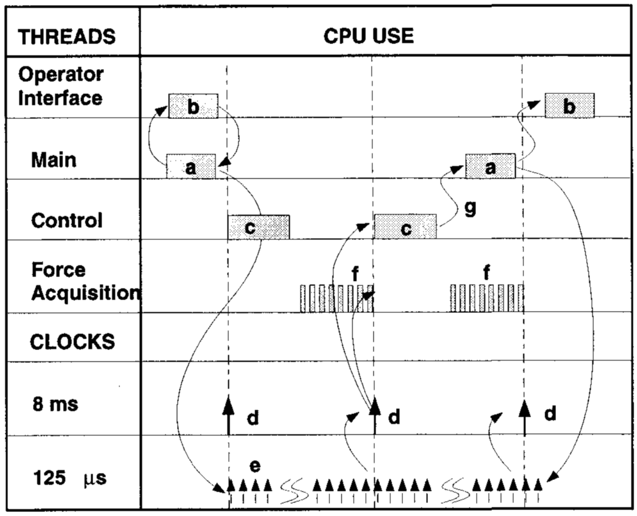


**Figure 3:** Diagram of the software timing

execute the position control loop at a fix interval $T = 896\,\mu s$, but they update their set point at a variable interval which can be set at initialization time. The original PUMA controller sets this interval to $32 \times T$, which results in the $28\,ms$ position update cycle characteristic of the PUMA Mark I,II, and III controllers.

Following the direction provided in [11, 3, 6], and after examining the joint controller code, the moper program was modified to initialize the set point update period of the joint controllers to $8 \times T$, which produces a control cycle of approximately $8\,ms$. The next lower cycle of $4\,ms$ cannot be used with the current configuration, since data exchange and handshake signals between moper and the trajectory generator on the PC take approximately $4\,ms$ each set point update.

Recalling the proposed robot-aided procedure mentioned in Section 1, one can think of the robot as functioning in two different modes. During the initial phase, the surgeon holds the needle by hand and moves the robot as if it were a passive mechanical device. Here, the robot is under *position accommodation* control, and the forces sensed at the wrist modify the robot position set point, making the robot follow the surgeon hand. During the needle insertion, the robot is moved under a quasi-hybrid control, in which one of the axis of the task reference frame is controlled in compliance mode, and the other axes are in position control mode. This hybrid controller differs from similar controllers [12], since no explicit force control is used. This variation is necessary to ensure that both position and force control set points are not exceeded during needle insertion.

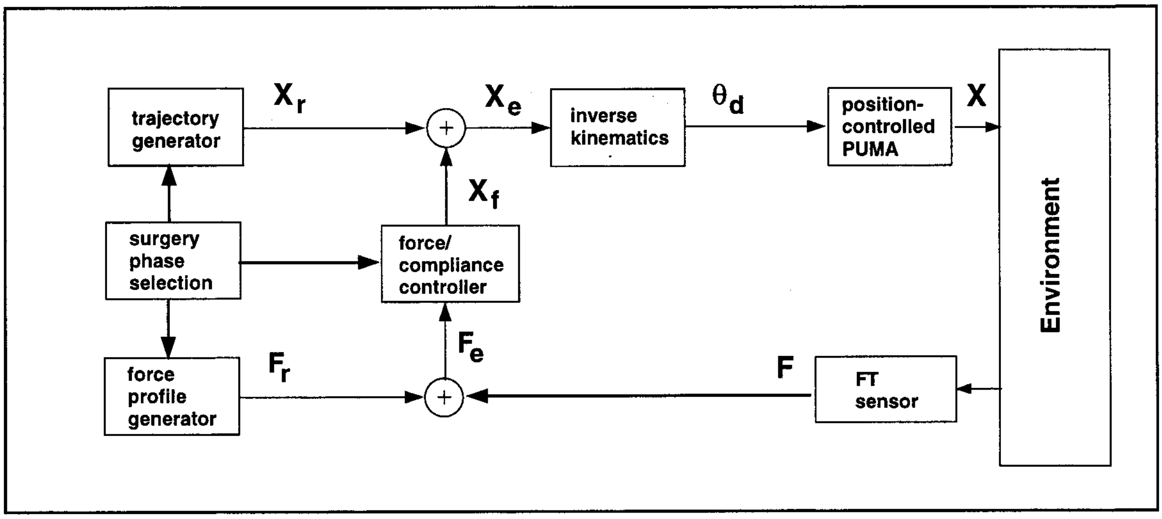


**Figure 4:** Block diagram of the control system.

The hybrid controller operates in task coordinate frame, which ensures that the surgical tool does not exceed the maximum forces allowed. The characteristic of the robot trajectory during needle insertion justifies the use of the hybrid controller. In this case in fact, the task frame in which compliance and position control are decomposed remains constant during needle insertion. The force control direction, i.e. the approach vector of the robot motion corresponding to the needle axis, is also constant in the task frame, and therefore the decomposition of the force feedback with respect to the robot trajectory can be carried out in advance before starting the motion.

During the first phase of the procedure, the robot is controlled in *position accommodation* mode, and the force error $F_e(s) = F_r - F(s)$ is used as the position command input. The reference force is $F_r = 0$ and the force controller $K_f(s)$ generates the necessary control action so that the robot motion zeros the sensed forces $F(s)$. The expression for $K(s)$ is given by:

$$K_f(s) = k_p + \frac{k_i}{s + \tau_f} \qquad (1)$$

where $k_p$ and $k_i$ are the proportional and integral force feedback gains, respectively, and $\tau_f$ is a suitable filter constant.

During needle insertion the robot is controlled by the hybrid controller which guarantees that the needle does not exceed the maximum force threshold and tracks the prescribed trajectory. The position controller relies on the PUMA joint servo controllers to ensure good trajectory tracking. The position set points are computed using the cycloidal interpolation given by the following equations:

$$\omega = \frac{2.0\pi}{T} \qquad (2)$$

$$s = \frac{\omega t - \sin(\omega t)}{2.0\pi} \qquad (3)$$

$$\dot{s} = \frac{\omega(1 - \cos(\omega t))}{2.0\pi} \qquad (4)$$

$$\ddot{s} = \frac{\sin(\omega t)\omega^2}{2.0\pi} \qquad (5)$$

where $s$, $\dot{s}$ and $\ddot{s}$ are respectively the value of the distance parameter on the trajectory, its first time derivative, its second time derivative, $T$ is the total motion time, and $t$ is time. This interpolation ensures that the end points of the trajectory have zero velocity and acceleration.

The compliance part of the hybrid controller is conceptually identical to the force controller, and is implemented as an outer feedback loop around the inner position controller. Here, force feedback is used to reduce the stiffness of the position controller when the needle reaches an obstacle. This allows the feedback forces $F(s)$ to alter the the reference position $X_r(s)$, and eventually stop the needle. The compliance controller used is given by the following equations:

$$K_c(s) = k_1 + \frac{k_2}{s + \tau_c} \qquad (6)$$

where $k_1$ and $k_2$ are the controller gains, and $\tau_c$ is a suitable filter constant. Under compliance control, the manipulator behaves as a spring with apparent stiffness determined by the controller gains.

Figure 3 shows the block diagram of the position and hybrid control schemes. The initial selection of the surgery phase determines the control type and the gains of the controllers. Then the trajectory generator produces constant set points, during the initial force-controlled phase, or send the set points of the needle insertion phase to the PUMA. The

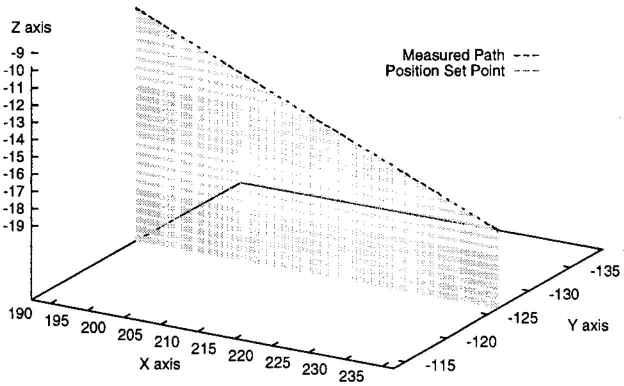


**Figure 5:** Plot of the position control experiment.

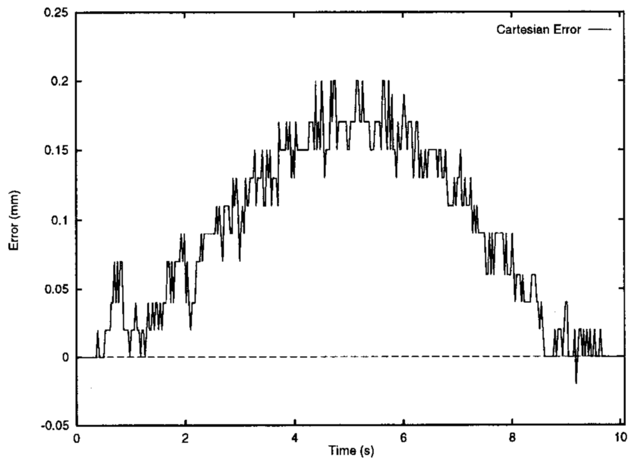


**Figure 6:** Plot of the tracking error.

force/compliance controller modifies the position set points to account for the feedback from the force sensor.

## 4. Experimental Results

In this section, we report on our current tests with the PUMA 260 manipulator used under position and force control modes.

The plot of a typical position control experiment is shown in Figures 4 and 5. The experiment consists of moving the robot arm along a straight line trajectory, from the initial position ($x = 187.39\,mm$, $y = -126.24\,mm$, $z = -9.81\,mm$), to the final position ($x = 236.56\,mm$, $y = -126.24\,mm$, $z = -18.76\,mm$), with constant needle orientation.

The purpose of these tests is primarily to evaluate the effects of the trajectory generation and communication protocol on the tracking performance of the arm. Figure 5 shows that the arm tracks the cycloidal Cartesian motion trajectories accurately. The maximum tracking error for this experiment is of $0.2\,mm$, and is characteristic of the trajectories in the range

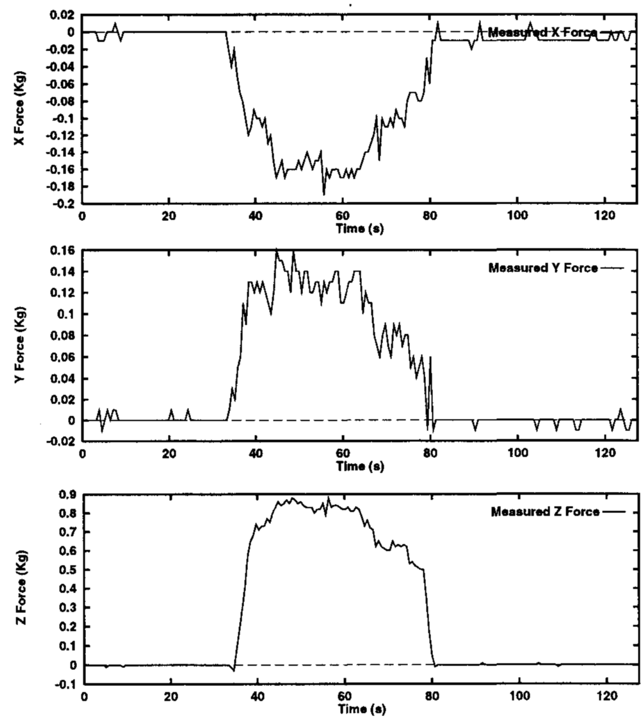


**Figure 7:** Plot of the forces during manual motion.

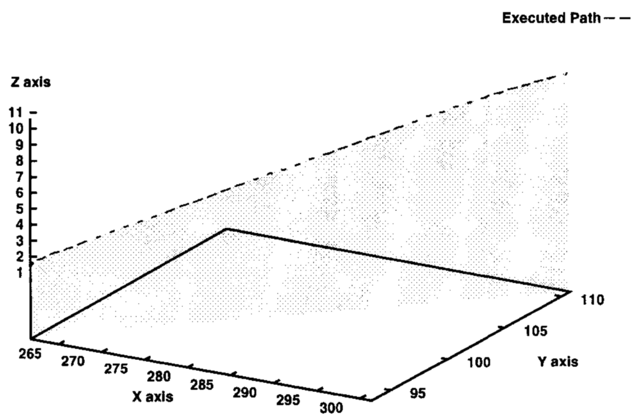


**Figure 8:** Plot of the manual motion.

10 to $15\,cm$ traversed at a velocity of 3 to $10\,mm/s$, typical of discectomy procedures.

Figures 6 and 7 show an example of the first phase of the surgical procedure, during which an operator moves the PUMA arm by hand, holding the needle as if it were a surgical instrument. Before the starting of the trajectory, the PUMA is positioned so that the Z axis of the wrist, corresponding to the needle, lies on a plane parallel to the XY world frame. The PUMA is then moved by *pulling* the needle in the positive direction of the Z axis, as shown by the values of the forces displayed in Figure 6. Figure 7 shows the plot of the motion whose primary component is in the positive Z direction of the wrist reference frame.

Since no control of the secondary motion directions is implemented yet, the plot shows small motion drifts in the X and Y directions, compatible with the X and Y forces shown in Figure 6. These secondary motions give a qualitative idea of the sensitivity of the force control, since a few grams applied to the needle are sufficient to move the PUMA arm.

## 5. Conclusions

A few significant enhancements to a robotic workstation for surgical procedures are described in this paper. The workstation is PC-based and uses a PUMA 260 manipulator controlled by algorithms located in the PC. The enhancements described consist of the development of a hybrid position/compliance controller for the manipulator, which was made possible by the reduction of the control period of the PUMA manipulator. The force and hybrid controllers are motivated by the requirements of surgical procedure for which the system is being developed. The force control is used during the initial phase of a percutaneous discectomy, when the surgeon uses the robot as a manual instrument to position the tip of a surgical needle on the patient body. The hybrid controller is used during the second phase of the procedure, when the medical personnel is removed from the patient and monitors remotely the position of the needle through fluoroscopic images.

The experiments carried out so far show that the PC-based controller is able to handle successfully the position and force control of PUMA with submillimeter position errors. Force experiments have shown good sensitivity to manual input, and the capability of precisely follow the operator hand motions.

In the future, we plan to test extensively the surgical workstation during realistic operating room experiments. We also plan to start experiments of force control on biological tissues, to define the requirements of robotic systems for these applications. We will also improve the overall safety of the system, by developing appropriate calibration procedures, force monitoring when using the flexible surgical needle, kinematic analysis of trajectory singularities, and external sensors.

## 6. Acknowledgment

The research described in this paper has been carried out at the Istituti Ortopedici Rizzoli. Part of the work has been done at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

## References

[1] Anonymous. Controlling the PUMA robot arm without using VAL. Technical report, Unimation Inc.

[2] C. Casadei, P. Fiorini, S. Martelli, M. Montanari, and A. Morri. A PC-based Workstation for Robotic Discectomy. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.

[3] P.I. Corke. The Unimation PUMA Servo System. Technical Report MTM-226, CSIRO, Automaiton Group, July 1994.

[4] P. Fiorini, H. Seraji, and M. Long. A pc-based controller for dexterous manipulators. *IEEE Robotics and Automation Magazine*, pages 1–9, September 1997.

[5] V. Hayward and J. Lloyd. RCCL user's guide. Technical report, McGill Research Centre for Intelligent Machines, Montreal, Quebec, Canada, 1992.

[6] J. Lloyd. Personal communication.

[7] John Lloyd. RCCL/RCI hardware installation notes. Technical report, McGill Research Centre for Intelligent Machines, Montreal, Quebec, Canada, 1992.

[8] John Lloyd. RCCL/RCI startup and installation guide. Technical report, McGill Research Centre for Intelligent Machines, Montreal, Quebec, Canada, 1992.

[9] S. Martelli, M. Di Silvestre, P. Dario, M. Fadda, M. Marcacci, and S. Giannini. A Proposal for a semi-automatic procedure for percutaneous discectomy. In *International Conference on Advanced Mechatronics (ICAM'93)*, Tokyo, Japan, November 1-3 1993.

[10] J. Mills et al. Development of a robot control test platform. *IEEE Robotics and Automation Magazine*, 2(4):21–28, December 1995.

[11] R. Vistnes. Breaking away from VAL, or how to use your PUMA without using VAL. Technical report, Unimation Inc., 1981.

[12] S. Yoshikawa. *Foundations of Robotics*. MIT Press, Boston, MA, 1990.